



مقدمه

این تمرین برای آشنایی با برنامه‌نویسی بازگشتی طراحی شده است. توجه کنید که پرسش‌ها حتماً باید به روش بازگشتی حل شوند، هر چند ممکن است روش‌های دیگری نیز برای حل آن‌ها وجود داشته باشد.

پرسش‌ها

۱. رمز پوچی‌ها

شرح مسئله

پوچی؛ نامی که انسان‌ها برای موجودات بیگانه‌ای که به تازگی به سطح کره زمین آمدند برگزیده‌اند. این موجودات با وجود تاریکی و پوچی ظاهریشان دارای حیاتی هوشمند هستند. تعدادی از این موجودات برای نجات هموعان خود که توسط گروهی از انسان‌ها به اسارت گرفته شده‌اند با تعدادی از انسان‌ها دست به همکاری زده‌اند.



به منظور فاش‌نشدن پیام‌هایی که بین این موجودات و انسان‌هایی که با آن‌ها همکاری می‌کنند رد و بدل می‌شود، کلمات آن‌ها رمزگذاری می‌شوند به شکلی که برای رمزگذاری یک رشته حرف میانی آن در نظر گرفته می‌شود. حرف میانی یک رشته به ازای عدد صحیح k برای رشته به طول مثبت $2k - 1$ و $2k$ برابر حرف k ام آن رشته است. در صورتی که در هر دو طرف حرف میانی رشته، زیررشته وجود داشته باشد حرف میانی در ابتدا نوشته می‌شود و سپس نتیجه رمزگذاری زیررشته سمت چپ و پس از آن نتیجه رمزگذاری زیررشته سمت راست نوشته

می‌شود. دقت داشته باشید برای هر رشته عملیات رمزگذاری تا وقتی که دو زیررشته سمت چپ و راست در هر دو طرف حرف میانی آن وجود داشته باشد ادامه پیدا می‌کند. حال از شما خواسته شده برنامه‌ای بنویسید که به انسان‌ها در رمزگذاری کلمات کمک کند.

ورودی

در تنها خط ورودی، رشته s که باید رمزگذاری شود ورودی داده می‌شود.

خروجی

در تنها خط خروجی، نتیجه رمزگذاری رشته ورودی نمایش داده می‌شود.

محدودیت‌ها

$$1 \leq |s| \leq 10^6 \quad \bullet$$

ورودی و خروجی نمونه

ورودی	خروجی
code	ocde
Cipher	pCiehr
acknowledgement	encakwolmgdenet

شرح ورودی و خروجی نمونه

کلمه Cipher را در نظر می‌گیریم. حرف میانی آن p ، زیررشته سمت چپ حرف میانی آن ci و زیررشته سمت راست حرف میانی آن her است. حرف p نوشته می‌شود. زیررشته سمت چپ بررسی می‌شود که در آن حرف میانی C است اما زیررشته سمت چپ ندارد پس ci نوشته می‌شود. پس از بررسی زیررشته سمت چپ، زیررشته سمت راست که کلمه her است بررسی می‌شود که در آن حرف میانی e ، زیررشته سمت چپ حرف میانی h و زیررشته سمت راست حرف میانی r است. حرف e نوشته می‌شود. سپس زیررشته سمت چپ بررسی می‌شود که کلمه h است و حرف میانی آن خود h است و زیررشته سمت چپ و راست ندارد پس حرف h نوشته می‌شود سپس زیررشته سمت راست که برابر r است بررسی می‌گردد، حرف میانی آن r است و زیررشته سمت چپ و راست ندارد پس r نوشته می‌شود.

۲. مراسم پوچی‌ها

شرح مسئله

پوچی‌ها می‌خواهند برای سال نو مراسم مخصوص به خودشان را برگزار کنند. مراسم آن‌ها به این صورت است که در یک صف به صورت تصادفی می‌ایستند. سپس هر کدام کارت تبریکی که از قبل آماده کرده است را به بلندقدترین فرد از افرادی که در سمت راست او قرار دارند می‌دهد. اما از زمانی که پوچی‌ها به زمین آمده‌اند بینایشان ضعیف شده است. به همین خاطر از شما خواسته‌اند تا به آنها کمک کند تا هر کس بلندقدترین فرد در سمت راست خود را پیدا کند.

برداری از قد پوچی‌ها به عنوان ورودی داده شده‌است. می‌خواهیم به ازای هر نفر بلندقدترین پوچی سمت راست آن را پیدا کنیم. بدین منظور دو پیاده‌سازی به همراه صورت پروژه در دو فایل جداگانه در اختیار شما قرار داده‌شده تا کد توابع فایل‌ها را کامل کنید. توجه کنید امضای توابع را تغییر ندهید یا تابع جدیدی اضافه نکنید. همچنین دقت داشته باشید تابع خواندن ورودی، عناصر وکتور نتیجه را به همان ترتیب ظاهر شدن عناصر ورودی برمی‌گرداند و تابع نمایش خروجی عناصر را به همان ترتیبی که در وکتور آمده است در خروجی نمایش می‌دهد. برای هریک از پیاده‌سازی‌ها یک متن راهنمایی در زیر آمده‌است.

۲.۱. پیاده‌سازی اول

به ازای هر عدد در ورودی یک struct در نظر گرفته و در آن مقدار عدد و مقدار بزرگترین عدد سمت راستش را ذخیره می‌کنیم. در ابتدا مقدار بزرگترین عدد سمت راست هر عدد را ۱- در نظر می‌گیریم. برداری از این struct ها برای اعداد ورودی می‌سازیم. سپس برای هر یک از اعداد بزرگترین عدد سمت راستش را پیدا کرده و در struct مربوط به آن ذخیره می‌کنیم.

سوال: مشکل پیاده‌سازی اول چیست؟ استفاده از پیاده‌سازی دوم چه مزیتی دارد؟

پاسخ: معمولاً رفتاری که از یک تابع در زمینه ورودی‌ها و خروجی انتظار می‌رود دریافت آرگومان‌ها و بازگرداندن نتیجه از طریق مقدار بازگشتی است و انتظار نمی‌رود مقادیر آرگومان‌های ورودی یک تابع به واسطه آدرس آن‌ها توسط تابع تغییر کند. تغییر مقادیر آرگومان‌های تابع به این شکل موجب کم‌شدن خوانایی و فهم کد می‌شود و راه را برای تولید باگ‌های متنوع در برنامه باز می‌کنند. در مواقعی که امکان تغییر این روند وجود دارد بهتر است این روش استفاده اصلاح شود. در پیاده‌سازی دوم این اشکال پیاده‌سازی اول اصلاح شده است.

۲.۲. پیاده‌سازی دوم

برداری از اعداد ورودی می‌سازیم. بردار جدیدی برای ذخیره بزرگترین عدد سمت راست هر عدد تعریف می‌کنیم. برای هر عدد بزرگترین عدد سمت راستش را پیدا کرده و به بردار اضافه می‌کنیم.

ورودی

در تنها خط ورودی، قد پوچی‌ها در صف داده می‌شود.

خروجی

در تنها خط خروجی، قد بلندقدترین فرد سمت راست هر پوچی چاپ می‌شود.

ورودی و خروجی نمونه

ورودی	خروجی
1 2 3 4	4 4 4 -1
2 1 6 3	6 6 3 -1
5 5 5 3 1 2	5 5 3 2 2 -1

شرح ورودی و خروجی نمونه

برای ورودی نمونه دوم داریم:

سمت راست نفر چهارم صف (با قد 2) سه نفر با قد های ۱ و ۶ و ۳ ایستاده‌اند که بلندقدترین آنها قد 6 است.
سمت راست نفر سوم صف (با قد 1) دو نفر با قد های ۶ و ۳ ایستاده‌اند که بلندقدترین آنها قد 6 است.
سمت راست نفر دوم صف (با قد 6) یک نفر با قد ۳ ایستاده‌است. پس بلندترین قد، قد 3 است.
سمت راست نفر اول صف (با قد 3) کسی ایستاده‌است. پس بلندترین قدی وجود نداشته و 1- می‌نویسیم.

۳. جدول پوچی‌ها

شرح مسئله

مدتی پس از کار با روش رمزنگاری‌ای که در سوال اول به آن پرداختید، پوچی‌ها متوجه ضعیف بودن این روش در ارتباط با انسان‌ها شدند در نتیجه تصمیم گرفتند کلمات کلیدی و مهم خود را از روش دیگری به انسان‌هایی که با آن‌ها همکاری می‌کنند برسانند. آن‌ها تصمیم گرفتند کلمات کلیدی خود را در جدول کلماتی که هر هفته در بخش پازل روزنامه چاپ می‌شود بگنجانند.

R	A	D	I	A	T	O	R	I	N	T	E	R	C	O	O	L	E	R	N	O	T	S	I	P
O	I	G	N	I	T	I	O	N	C	O	I	L	T	U	R	B	O	C	H	A	R	G	E	R
C	O	N	N	E	C	T	I	N	G	R	O	D	T	H	R	O	T	T	L	E	B	O	D	Y
K	H	E	G	G	U	L	P	K	R	A	P	S	D	A	E	H	R	E	D	N	I	L	Y	C
E	O	S	L	S	C	R	A	N	K	S	H	A	F	T	W	A	T	E	R	P	U	M	P	L
R	I	T	E	B	A	L	A	N	C	E	S	H	A	F	T	C	A	M	S	H	A	F	T	O
A	L	A	V	L	M	O	T	O	R	M	O	U	N	T	C	R	A	N	K	C	A	S	E	L
R	P	R	L	O	F	F	U	E	L	I	N	J	E	C	T	O	R	M	A	N	I	F	O	L
M	A	T	A	C	T	L	E	B	G	N	I	M	I	T	A	L	T	E	R	N	A	T	O	R
A	N	E	V	K	R	O	I	L	F	I	L	T	E	R	G	U	L	P	N	I	A	R	D	R
C	A	R	B	U	R	E	T	O	R	H	E	A	D	G	A	S	K	E	T	S	U	M	P	Y

یک کلمه در جدول وجود دارد به شرطی که با شروع از یک خانه جدول و حرکت به سمت هر خانه دلخواه در هشت طرف خانه فعلی کلمه ساخته شود به طوری که تنها یک بار از هر خانه در مسیر ساخت کلمه گذشته باشیم. حال از شما خواسته شده برنامه‌ای بنویسید که مشخص کند آیا کلمه‌ای خاص در جدول کلمات داده شده وجود دارد یا خیر و در صورتی که وجود دارد مسیر ساخت آن را نمایش دهد.

ورودی

ورودی مسئله کلمه‌ای که به دنبال آن هستیم و یک جدول مربعی شکل شامل حروف است. تضمین می‌شود کلمه داده شده حداکثر یک بار در جدول وجود داشته باشد.

در اولین خط ورودی رشته s که وجود آن در جدول باید بررسی شود به عنوان ورودی داده می‌شود. در دومین خط ورودی عدد صحیح n به عنوان طول جدول مربعی شکل به شما داده می‌شود. در هر یک از n خط بعدی n کاراکتر با فاصله از هم ورودی داده می‌شوند.

خروجی

در صورتی که کلمه ورودی در جدول وجود نداشت عبارت **No** در خروجی نمایش داده می‌شود، در غیر این صورت در اولین خط خروجی عبارت **Yes** و در n خط دیگر در هر خط n عدد با فاصله از هم نمایش داده

می‌شوند که مسیر پیمایشی که کلمه در آن پیدا شده را نمایش می‌دهند. کلمه یافت‌شده از خانه‌ای شروع شده و در مسیر خود خانه‌های دیگری را طی کرده است، نیاز است برای نمایش مسیر پیمایش در خروجی جدولی که نمایش می‌دهید برای خانه اول مسیر عدد ۱، خانه دوم عدد ۲ و به ترتیب تا آخر مسیر این اعداد قرار گرفته باشند. همچنین برای خانه‌هایی که در مسیر نیامده‌اند عدد ۰ نمایش داده می‌شود.

محدودیت‌ها

- $3 \leq n \leq 10$
- $\min(3, n) \leq |s| \leq n^2$

ورودی و خروجی نمونه

ورودی	خروجی
run 4 w u h z k v e r t n m c r y q l	No
secret 5 c t e i z m c r y g a d e g w y s p n o w b r t l	Yes 0 6 5 0 0 0 3 4 0 0 0 0 2 0 0 0 1 0 0 0 0 0 0 0 0

شرح ورودی و خروجی نمونه

برای ورودی نمونه دوم داریم:

c	t	e	i	z		0	6	5	0	0
m	c	r	y	g		0	3	4	0	0
a	d	e	g	w		0	0	2	0	0
y	s	p	n	o		0	1	0	0	0
w	b	r	t	l		0	0	0	0	0

نحوه تحویل

- کد مربوط به هر پرسش را به صورت یک پرونده¹ جداگانه و به ترتیب سوالات با نام‌های P1.cpp و P2-1.cpp و P2-2.cpp و P3.cpp بنویسید و سپس همه آن‌ها را به صورت یک پرونده فشرده با نام A2-SID.zip در صفحه eLearn درس بارگذاری کنید که SID شماره دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۸۹۹۹ باشد، نام پرونده شما باید A2-810198999.zip باشد که شامل پرونده‌های کدهای شما است.
- برنامه شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++11 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.
- استفاده از روش‌های بازگشتی، تمیزی کد، شکستن مرحله به مرحله مسئله و طراحی مناسب در کنار تولید خروجی دقیق و درست، بخش مهمی از نمره شما را تعیین خواهد کرد.
- درستی برنامه شما از طریق آزمون‌های خودکار سنجیده می‌شود؛ بنابراین پیشنهاد می‌شود که با استفاده از ابزارهایی مانند diff خروجی برنامه خود را با خروجی‌هایی که در اختیارتان قرار داده شده است مطابقت دهید. مطلبی در راستای آشنایی بیشتر شما با همین موضوع در صفحه‌ی درس بارگذاری می‌شود.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.

¹ File